Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 151 (2003) 447–460
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Invited Review

Nurse rostering problems—a bibliographic survey

B. Cheang [a], H. Li [b], A. Lim [c,*], B. Rodrigues [d]

[a] IOPT, Incubation Centre, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore
[b] NJIT-CCS, Department of Computer Science, GITC 4400, University Heights, Newark, NJ 07102, USA
[c] Department of IEEM, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[d] School of Business, Singapore Management University, 469 Bukit Timah Road, Singapore 259756, Singapore

Received 7 March 2002; accepted 18 December 2002

Abstract

Hospitals need to repeatedly produce duty rosters for its nursing staff. The good scheduling of nurses has impact on the quality of health care, the recruitment of nurses, the development of budgets and other nursing functions. The nurse rostering problem (NRP) has been the subject of much study. This paper presents a brief overview, in the form of a bibliographic survey, of the many models and methodologies available to solve the NRP.

Keywords: Manpower rostering; Scheduling

1. Introduction

Every hospital needs to repeatedly produce duty rosters for its nursing staff. Properly scheduling the nursing staff has a great impact on the quality of health care [69], the recruitment of nursing personnel, the development of a nursing budgets and various other functions of the nursing service. Duty rosters can be generated manually by nursing officers for each hospital unit. However, scheduling nurses has always been difficult. A general overview can be found in [39,76]. The main reason lies in that hospitals need to be staffed 24 hours a day over seven days a week. In addition, in many hospitals, nurses are allowed to request preset shifts, while other nurses are scheduled around these pre-set shifts. Usually, nursing officers spend a substantial amount of time developing rosters especially when there are many staff requests, and where even more time can be consumed in handling ad hoc changes to current duty rosters. Because of tedious and time-consuming manual scheduling, and for various other reasons, the nurse rostering problem (NRP) or the nurse scheduling problem (NSP) has attracted much research attention.

The NRP involves producing a periodic (weekly, fortnightly, or monthly) duty roster for nursing staff, subject to a variety of hard/soft constraints such as legal regulations, personnel policies, nurses' preferences and many other

* Corresponding author. Address: School of Computing, Department of Computer Science, The National University of Singapore, 3 Science Drive, Singapore 117543, Singapore. Tel.: +65-874-6891; fax: +65-779-4580.
E-mail address: iealim@ust.hk (A. Lim).

0377-2217/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/S0377-2217(03)00021-3

requirements that may be hospital-specific. These constraints can vary from one hospital to another while the objectives in rostering can also vary. These have resulted in a whole range of NRP models and, consequently, a wide range of solution approaches that have been developed for these models.

This paper presents a brief overview, in the form of a bibliographic survey, of the many models and methodologies available to solve the NRP. In Section 2 we give a brief overview of the modelling of the NRP and in Section 3, we describe the solution approaches that are available for this problem. In Section 4, we discuss the evaluation of some of these approaches and in Section 5 we provide some conclusions to this survey and an extensive bibliography.

2. Modelling the NRP

2.1. Decision variables, parameters and domains

The NRP is commonly described by a nurse-day view, a nurse-task or nurse-time slot view and a nurse-shift pattern view.

A *nurse-day* view is a direct depiction of a two-dimensional duty rosters. Accordingly, the decision variables can be defined for each nurse on each day as v_{ij}, where $1 \leqslant i \leqslant N$ indexes the nurses and $1 \leqslant j \leqslant P$ indexes the days within a scheduling period. The domains of these variables consist of on-duty shifts and free shifts. On-duty shifts may include any number of shifts per day, but it is common to use only a morning shift (A) of eight working hours, an afternoon shift (P) of eight working hours, and a night shift (N) of eight working hours. Free shifts include day-off (O), compensation-off (CO), public holiday (PH), vacation leave (VL), study day (SD), maternity leave (ML), unpaid leave (UL), etc. Thus, the decision variables can typically take on 10 or more values, which increase computational efforts.

Heus and Weil [37] use a reduction of variable domains (see also, [2,10,44,46,60]). The idea is to set all values of the free shifts to 0. In the general situation, when there are Z shifts per day, v_{ij} can take $Z+1$ possible values:

$$v_{ij} = \begin{cases} 0 & \text{nurse } i \text{ is off duty on day } j, \\ 1 & \text{nurse } i \text{ works shift 1 on day } j, \\ \vdots & \\ Z & \text{nurse } i \text{ works shift } Z \text{ on day } j. \end{cases}$$

Their paper gives an example with three-shift day. The values of the free shifts are reduced to one value (F). There is only the morning shift (A), the afternoon shift (P) and the night shift (N), so that the decision variable will take on four possible values:

$$v_{ij} = \begin{cases} 0 & \text{nurse } i \text{ is off duty (F) on day } j, \\ 1 & \text{nurse } i \text{ works morning shift (A) on day } j, \\ 2 & \text{nurse } i \text{ works afternoon shift (P) on day } j, \\ 3 & \text{nurse } i \text{ works night shift (N) on day } j. \end{cases}$$

Table 1 shows part of a weekly roster which indicates the shifts allocated to the nurses, in a nurse-day view.

For 0–1 models, the decision variables can be customized to be v_{ijk}, where i, j are the same indexes as that for v_{ij}, and $1 \leqslant k \leqslant Z$ indexes the Z possible shifts in a day. In the above example, for $Z=3$, v_{ijk} is binary:

$$v_{ijk} = \begin{cases} 1 & \text{nurse } i \text{ works shift } k \text{ on day } j, \\ 0 & \text{otherwise.} \end{cases}$$

Kragelund and Kabel [48], for example, used this representation (see also [60]). Both v_{ij} and v_{ijk} are nurse-day view representations, and can be used a decision variables to model the NRP problem.

A *nurse-task* view is a close variant of the nurse-day view. The decision variable can be defined for each variable in each shift as v_{is}, where $1 \leqslant i \leqslant N$ indexes the nurses and $1 \leqslant s \leqslant Z$ indexes tasks within a scheduling period. The only difference between nurse-task view and nurse-day view is that the shift defined in nurse-task view may not necessarily correspond to a "day". Jaumard et al. [44] (see also, [22,84]) proposed binary models with:

Table 1
Nurse-day view

Nurse ID	Mon	Tue	Wed	Thu	Fri	Sat	Sun
N-01	A	P	N	A	F	F	P
N-02	P	A	N	P	F	F	A
N-03	N	N	N	F	F	F	A

$$v_{is} = \begin{cases} \text{used(scheduled)} & \text{nurse } i \text{ receives task } s, \\ \text{idle(free)} & \text{otherwise.} \end{cases}$$

A *nurse-shift pattern* view is different from the above two views. As staff can prefer to have simple shift schedules, it is desirable to have less shift patterns. Aickelin [6] (see also, [32,55]) promulgates his problem as an IP and sets decision variables as v_{ip}, which $1 \leqslant i \leqslant N$ indexes nurses and $1 \leqslant p \leqslant M$ indexes shift patterns where:

$$v_{ip} = \begin{cases} 1 & \text{nurse } i \text{ works shift pattern } p, \\ 0 & \text{otherwise.} \end{cases}$$

Typically, parameters in the NRP would include the following, for example: working shifts per week if night shifts are worked, preference costs of particular nurses working on particular shift pattern, working shifts per schedule if day shifts are worked, working shifts per schedule if both day and night shifts are worked, demand for certain grade of nurses on day and on night shifts.

2.2. Constraints

Constraints that commonly occur with NRPs can be divided into two classes, generally: hard constraints and soft constraints—as is the case in other types of problems. Hard constraints usually include coverage requirements (for example, staff demand per day per shift type per skill category) while soft constraints are usually those involved with time requirements on personal schedules. The goal is always to schedule resources to meet the hard constraints while aiming at a high quality result with respect to soft constraints. Commonly occurring constraints are listed below:

1. Nurses workload (minimum/maximum).
2. Consecutive same working shift (minimum/maximum/exact number).
3. Consecutive working shift/days (minimum/maximum/exact number).
4. Nurse skill levels and categories.
5. Nurses' preferences or requirements.
6. Nurses free days (minimum/maximum/consecutive free days).
7. Free time between working shifts (minimum).
8. Shift type(s) assignments (maximum shift type, requirements for each shift types).
9. Holidays and vacations (predictable), e.g., bank holiday, annual leave.
10. Working weekend, e.g., complete weekend.
11. Constraints among groups/types of nurses, e.g., nurses not allowed to work together or nurses who must work together.
12. Shift patterns.
13. Historical record, e.g., previous assignments.
14. Other requirements in a shorter or longer time period other than the planning time period, e.g., every day in a shift must be assigned.
15. Constraints among shifts, e.g., shifts cannot be assigned to a person at the same time.
16. Requirements of (different types of) nurses or staff demand for any shift (minimum/maximum/exact number).

Table 2 lists some papers which have one or more of the constraints described above.

From Table 2, we see that constraints 1, 3, 5, 6, 7, 8, 10, 14 and 16 are common in NRPs. In particular, we note that constraint 16 must be covered in any solution.

The characteristics of a NRP is that they are usually highly constrained and often over-constrained, are usually constrained by personnel preferences and priorities (cyclical schedules), and

Table 2
Work done where constraints occur

Constraint type	References
1	[2,21,24,28,47,64,78]
2	[28,26,78]
3	[16,23,24,26,47,78,81]
4	[28,46,78,83]
5	[2,16,46,47,78,81,83]
6	[16,21,24,64,78]
7	[2,21,24,28,47,78]
8	[21,26,28,46,47,78,81]
9	[26,47,78]
10	[16,21,23,24,26,64,78,81]
11	[46,78]
12	[21,28,78]
13	[2,28,78,83]
14	[21,23,26,46,78]
15	[26,46]
16	[2,16,21,23,24,26,28,46,47,64,78,83]

the problem has been tackled manually in practice. Other soft constraints can include having identical shift types on the weekend, balance in workload, assigning complete weekends and patterns enabling specific cyclic constraints.

2.3. Objective functions

Typically, with optimization problems (OPs) we find models that use standard objective functions, such as those for mathematical programming (MP) models. In other models, we find target or evaluation functions that are used to guide the generation of results or to evaluate results. In [6,32], for example, we find the objective $\sum_{i=1}^{n}\sum_{j\in F(i)} p_{ij}x_{ij} \rightarrow \min!$, where p_{ij} is the penalty cost of nurse i working on shift pattern j, x_{ij} is the decision variable with a nurse-shift view and $F(i)$ is the set of feasible shift patterns for nurse i, where the purpose is to minimize the total penalty cost for all nurses. This is subject to the constraints that each nurse works exactly one feasible shift pattern and a demand for nurses is fulfilled for every grade for every day and night. In other situations, a penalty function approach can be used when feasibility cannot be guaranteed. In [7] we find such a function where the penalty is proportional to the number of uncovered shifts for the problem and is used to evaluate the fitness of solutions in a GA context. In [60], we find, for example, a function that will minimize the cost of schedules and the penalty for violating shift balance. These functions vary in complexity depending on the problem at hand, and can be simple or as complex as that found in [66], for example.

2.4. Problem types

Depending on the models and constraints, a NRP, generally, can be classified as a OP or as a decision problem.

2.4.1. Optimization-type problems

In much of the early work, the NRP was treated as an OP. Using MP, the problem was formulated to minimize or maximize an objective function. In the case where there were multiple goals with priorities, goal programming (GP) and other tools were used. MP is an exact approach to combinatorial OPs. Traditional methods from linear programming, integer programming, GP, networks have been employed to solve the NRP (see, for example, [17,44,52,61,65,79,83,84]). In many instances, however, the NRP, like many other problems, can have too many constraints to allow for a MP formulation.

2.4.2. Decision-type problems

In situations where there are a large number of constraints to be dealt with, it can be more appropriate to model the NRP as a *constraint satisfaction problem* (CSP). A CSP can be defined as a three-tuple (V, D, C), where, V is a set of n variables, v_i, $i = 1, \dots, n$; D is a set of n domains, D_i, $i = 1, \dots, n$, such that each D_i is the finite set of possible values for each v_i; C is a finite set of constraints, each of which acts on a subset of variables in V restricting the possible combinations of values that these variables can take.

Feasible solutions to the CSP are the assignments of values to variables satisfying all constraints. Methodologies for solving CSP (see [12,49]) have been extensively studied. Solutions to CSP can be found by systematic tree-search such as back-tracking (BT). However, BT has the inherent drawbacks of thrashing and redundant constraint checks. Thrashing, which leads to repeated failure, can be avoided by using strategies such as back-jumping (BJ) which is applied directly to the variable causing the failure. Backmarking (BM) aims at eliminating redundant constraint checks by preventing the same constraint from being tested repeatedly. In contrast to BJ and BM, more attention is paid to overcoming these drawbacks in constraint propagation which removes inconsistent values from variables' domains until the solution is obtained or failure is reported when the domain of any variable becomes empty. However, the checking of simple constraint consistency, such as node consistency and arc consistency, is not sufficient to produce a solution without a search procedure. In practice, combinations of BT and constraint propagation are a common way of solving CSP. Methods, such as forward checking (FC) and maintaining arc consistency (MAC), usually use constraint propagation algorithms to simplify problems, and then use BT to obtain solutions. In addition, incorporating variable-ordering or value-ordering heuristics into these hybrid approaches can greatly improve search speed.

In the last few years, constraint programming (CP) has received high attention because of its potential for solving difficult problems. Currently, there is a variety of constraint logic programming languages (see, for example, [42,43]) and packages for conventional programming languages, such as Prolog III [29], CLP(R) [43], CHIP [31,82], and ILOG SOLVER Package [40] for C++. These languages and packages provide for expressive and flexible problem specification, allowing quick program development for hard problems. Despite its short history, CP has been applied widely to solve hard problems including CSPs.

Other approaches for solving CSP include heuristics and meta-heuristics such as hill climbing (HC), tabu search (TS), genetic algorithms (GA) and simulated annealing (SA).

2.4.3. Constraint optimization problems

In many real-life applications, we usually do not seek any solution, but rather a good solution. The quality of solutions is measured by single or multiple criteria which are usually incorporated into an objective function. The goal is to find a solution which maximizes or minimizes the value of the objective function. A problem modeled this way is referred to as a *constraint satisfaction optimization problem* (CSOP), which is a problem that consists of a standard CSP together with an objective function.

For a CSOP, well-known Branch and Bound (B&B) algorithms (see [53], for example) can be used to find optimal solutions. B&B needs an evaluation function to map a partial labeling of decision variables to a numerical value, which represents an underestimate (in case of minimization) for the best complete labeling from the partial labeling. If this value exceeds a given bound, which records the value of the current best solution, the subtree under the current partial labeling is pruned. The efficiency of B&B is determined by the quality of the evaluation function and whether a good bound is found early. The combination of B&B with CP can improve search speed.

Many NRPs are over-constrained, so that to find assignments to decision variables without violating any constraints is usually impossible. Consequently, the problem specification has to provide for the relative importance of constraints so that a solution to such a problem is allowed to violate a few constraints according to a priority order of constraints. Naturally, such a NRP can be modeled as a *partial constraint satisfaction problem* (PCSP) (see [34]), which consists of a standard CSP and an objective function, as does CSOP. However, PCSP differs from CSP in that PCSP does not require that every constraint be satisfied. In this sense, PCSP can be viewed as a generalization of CSOP.

Constraint hierarchies (CH) by Borning and coworkers [18,19] is another approach for handling over-constrained problems. In CH, the constraints are weakened explicitly by specifying their hierarchical levels. For constraints with the same hierarchical level, the importance of constraints is further specified by weight factors. The hierarchical structure of constraints do not allow the weakest constraint to influence the result at the expense of not satisfying a stronger constraint. In this sense, CH is a special class of PCSP. The *hierarchical constraint satisfaction problem* (HCSP) by Meyer auf'm Hofe [57,58] was derived from CH. CSP algorithms, B&B algorithms, and combinations of AI approaches with B&B algorithms can be customized to solve the PCSP and the CH/HCSP.

3. Solution approaches to the NRP

3.1. Initialization, pre- and post-planning options

In models that require an initial feasible schedule satisfying hard constraints, the choices can include: the empty schedule, the previous planning period when requirements and constraints are similar or the current schedule when the requirements have changed. Pre-planning and post-planning the NRP are always options and part of sensitivity analysis. For pre-planning, for example, it is possible to set hard constraints to preferred requirements and minimum requirements and in post-planning, it is possible add shift types to preferred requirements. Generally, in solving the NRP, there can be quick approach, where the aim is to generate an acceptable schedule while there can also be more thorough approaches depending on the problem and the needs of the hospital. Moreover, nurse rostering should be balanced against sensitivity to changes, since hospitals are very dynamic environments. Optimal solutions derived from techniques with high computing times are usually less valuable than one that is based on an flexible algorithm or user intuitive application.

3.2. Solution approaches

In general, there are two basic types of scheduling used for the NRP, which are cyclic and non-cyclic scheduling. In cyclic scheduling, each nurse works in a pattern which is repeated in consecutive scheduling periods, whereas, in non-cyclic scheduling, a new schedule is generated for each scheduling period. Cyclic scheduling was first used in the early 1970s due to its low computational requirements and the possibility for manual solution. The algorithms for the NRP, generally, deal with either cyclic scheduling or non-cyclic scheduling.

In the past decades, many approaches have been proposed to solve NRPs as they are manifested in the different models. The three commonly used general methods are MP, heuristics and AI approaches. Most heuristic approaches focus on solving cyclic scheduling problems, while MP and AI approaches can be found to be used on both cyclic and non-cyclic problems.

Solution approaches for the NRP can be classified into two main categories: The optimization approach and the decision approach. The optimization approach is usually based on MP techniques, while the decision approach usually employs heuristics and other AI tools.

3.3. Optimization—mathematical programming

Optimization approaches are usually based on MP. Some of the goals for optimization include: minimum staffing requirements, minimum desired staffing requirements, maximum satisfaction of nurses' preferences or their special requests, and so on. In general, optimization using MP can be classified in three categories: single-objective MP, multi-objective MP, and MP-based near-optimal approaches.

3.3.1. Single-objective MP

Single-objective MP involves maximizing a goal which is preferred by the decision-maker. Baker [11] proposed a cyclic schedule model which considers the case of two consecutive days off per week for each person. Bartholdi et al. [13] modeled the NRP as an IP with cyclic structured 0–1 constraint matrix. The IP was solved parametrically as a bounded series of network problems. Burns [23] used a cyclic model to study the case of 10 working days in a 14-day period with variable demands and alternate weekends off. Burns and Koop [24] considered cyclic assignments for a similar model with three workshift types and fixed cyclic specifications on working days and days off. Rosenbloom and Goertzen [74] presented an algorithm with three stages: generate a set of possible schedules which are seven-tuples of 0–1 depending on whether the day is off or on, formulate the problem as an IP, and produce a solution.

Beside cyclic models, non-cyclic models have also received much attention. Warner and Prawda [83] modeled the problem as a large-scale mixed-integer quadratic programming problem, to minimize a "shortage cost" of nursing care services for a period of three to four days subject to nursing skill class requirements, total personnel capacity constraints, integral assignment, minimum staffing requirements throughout the scheduling period and other relevant constraints. The problem is decomposed by a primal resource-directive approach into a 0–1 LP master problem, with smaller quadratic programming sub-problems.

Warner [84] posed a multiple-choice programming model which aims to maximize nurses' preferences, by considering the length of a work stretch, rotation patterns, requests for days off, and minimum numbers of nursing personnel of each skill class to be assigned to each day and a

4- to 6-week scheduling period. The problem is solved by a modification of Balintfy and Blackburn's algorithm for multiple-choice programming problems. In this two-phase algorithm, a specially designed nonlinear Phase I routine finds a feasible solution to meet various constraints, and a Phase II routine seeks to improve the Phase I solution by maximizing individual preferences for various schedule patterns while maintaining the Phase I solution.

Miller et al. [61] formulated the problem to minimize an objective function that balances the trade-off between staffing coverage and schedule preferences of individual nurses, subject to certain constraints on the nurses' schedules. The constraints are divided into hard and soft constraints. The hard constraints define sets of feasible nurse schedules, while violation of soft constraints results in a penalty cost that appears in the objective function. A coordinate descent algorithm was proposed to find near-optimal solutions.

Kostreva and Jennings [47] used MP to minimize the total aversion of all personnel to their schedules. The algorithms, based on Bender's decomposition, utilizes two alternating subproblems: generation of feasible sets of schedules and the optimal allocation of these schedules.

Millar and Kiragu [60] used a network model, which is in fact a shortest-path problem with side constraints, for cyclic and non-cyclic nurse scheduling with two workshift types. The model was solved using the CPLEX mixed-integer optimization software.

Jaumard et al. [44] presented a generalized 0–1 column generation model with a resource constrained shortest path auxiliary problem for nurse rostering. The master problem finds a configuration of individual schedules to satisfy the demand coverage constraints while minimizing salary costs and maximizing both nurse preferences and team balance. A feasible solution of the auxiliary problem is an acceptable schedule for a given nurse, with respect to collective agreement requirements such as seniority, workload, rotations and days off. A new resource structure was defined in the auxiliary problem in order to satisfy complex collective agreement rules specific to the problem.

3.3.2. Multi-objective approaches

Multi-objective models appear to be more realistic and are more flexible for weighting objectives by priority.

Berrada et al. [16] formulated the NRP as a multi-objective MP model. In this model, hard constraints must be satisfied, while soft constraints are treated as goals to be reached. The overall objective is to get as close as possible to these goals. Slack variables are introduced into the soft constraints, where the objectives are to minimize the values of these variables. Two different techniques, namely the sequential technique and the equivalent weights technique, were used to generate an efficient solution having the property that there is no other feasible solution that improves one of the objectives without worsening another one.

GP is better adapted for models with multi-objectives and priorities. Arthur and Ravidran [9] posed a four-goal (minimum staffing requirements, desired staffing requirements, nurses' preferences, and nurses' special requests) GP model which works in two phases. In the first phase, the nurses are assigned their day-on/day-off pattern for the two-week scheduling horizon by a GP model which allows for consideration of the multiple-conflicting objectives inherent in scheduling a nurse. The second phase makes specific shift assignment through the use of a heuristic procedure. The two-phase approach reduces the problem size considerably, thus reducing the computational effort. Musa and Saxena [64] and Ozkarahan and Bailey [70] also treated the NRP as GP models.

3.3.3. MP-based near-optimal methods

Inspired by Glover and McMillan [35], Valouxis and Housos [81] aimed to combine the strength of MP and AI approaches. The problem was formulated as an approximate IP model, where the IP problem is first solved and its solution further improved using TS.

Balakrishnan and Wong [10] used network model to solve the staff scheduling problem. In the model, a non-cyclic graph is defined, where the nodes represent the workshift or days off, while the arcs between nodes define the sequence of workshifts that form legal work stretches. A two-phase approach was used to obtain final duty rosters. Phase I applied a Lagrangian dual-based algorithm for determining a good lower bound for the problem, while Phase II used a k-shortest path approach to perform partial enumeration of paths in the network model and then to identify the solution paths.

3.4. Heuristic approaches

For combinatorial problems, exact optimization usually requires large computational times to produce optimal solutions. In contrast, heuristic approaches can produce satisfactory results in reasonably short times. In the recent years, meta-heuristics including, TS, GA and SA, have been proved to be very efficient in obtaining near-optimal solutions for a variety of hard combinatorial problems including the NRP.

3.4.1. Classical heuristics

Many heuristic approaches were straightforward automation of manual practices, which have been widely studied and documented in nursing administration literature (see [39,45], for example).

Basic heuristics can include, for example: *Shuffling* and *Greedy Shuffling*. In the first, the problem is solved for the worst schedule and then the quality is improved by exchanging a part of this schedule with a part from another person's schedule. Many human-inspired approaches can be found in Greedy Shuffling type algorithms which work by calculating all the shuffles for all personnel and listing them with the highest cost benefit first. This is repeated as many times as possible.

Howell [38] and Marchionno [56] described the necessary steps to develop cyclic schedules. Frances [33], Monroe [62], Mailer-Rothe and Wolfe [55], and Anzai and Miura [8] described computerized programs for producing cyclic duty rosters.

Ahuja and Sheppard [3] employed an interactive terminal facility to help decision-makers select work patterns to provide the needed coverage for given skilled nurse classes on each shift. Smith and Wiggins [77] presented a three-phase scheduling algorithm which first collects a summary of rostering data, then generates tentative shift schedules indicating shortages and averages in each unit, and finally manually adjusts the tentative shift schedules to produce final schedules. Randhawa and Sitompul [72] implemented a system using heuristics for pattern generation and pattern screening. Bell et al. [14] developed a visual interactive decision support system. The system used a heuristic to develop a basic pattern to meet shift and coverage constraints and to meet required staffing levels. Once the master pattern was set, the second and later weeks' schedules were derived from the first with modifications to fulfill requirements. The system provided an initial schedule to be shown to the decision-maker for necessary modifications. Okada and Okada [68] aimed at automating scheduling by following the manual method in a faithful manner. A system was implemented using Prolog which can describe various constraints with relative ease. In this system, shift assignments were determined on a day-to-day basis.

3.4.2. Meta-heuristics

TS approaches have been widely used to solve many combinatorial problems (see [36] for an overview of TS). Some TS approaches have been proposed to solve the NRP. TS is a search that moves iteratively from one solution to another by moves in a neighborhood space with the assistance of an adaptive memory. This memory forbids solution attribute changes recorded in the short-term memory to be reused. How long a restriction is in effect depends on the tabu tenure. In TS, a move, for example, can take on an assigned shift type from one nurse to another on the same day and a move not allowed (tabu) if, for example, the person does not belong to the skill category required or if there is already an assignment for that shift type. In TS, hard constraints remained fulfilled, while solutions move in the following way: calculate the best possible move which is not tabu, perform the move and add characteristics of the move to the tabu list. Dowsland [32] used TS with strategic oscillation to tackle the NRP in a large hospital. The objective is to ensure enough nurses are on duty at all times while taking account of individual preferences and requests for days off. The approach repeatedly oscillates between finding a feasible cover, and improving it in terms of preference costs. Nonobe and Ibaraki [66] proposed a tabu-based algorithm for the CSP as a foundation for a general problem solver. Experimental results were reported for several combinatorial problems including the NRP. Burke et al. [20] presented a hybrid TS approach that has been developed for a commercial nurse rostering system (Plane). In this approach, a feasible initial schedule is obtained using three possible strategies: (1) use current schedule when urgent changes in the schedule are required to avoid any drastic change of the schedule for other nurses; (2) use previous schedule when the constraints on the current and the previous planning period are similar; and (3) use random initialization for which the initial solution is then improved by a hybrid TS algorithms which combines TS with manual scheduling techniques to improve on results by making small changes manually. This is reported in [15].

GAs, which are stochastic meta-heuristics, have also been used to solve the NRP (see, for example, [4,5,7,59,73]). In GA, the basic idea is to find a genetic representation of the problem so that "characteristics" can be inherited. Starting with a population of randomly created solutions, better solutions are more likely to be selected for recombination into new solutions. In addition, new solutions may be formed by mutating or randomly changing old ones. For example, in the context of NRP, for crossover and mutation, the best personal schedule from each of the parents can be selected, a random selection from the personal schedule of parents can be selected, or we can select the best events in a schedule. Some of the best solutions in each generation are kept while others are replaced by newly formed solutions. Jan et al. [46] used GA for a problem with multiple criteria where the concept of a Pareto optimality scheme is used for the evaluation of the multi-criteria objective function. Aickelin and Dowsland [4,5] developed a GA approach to solve an NRP. Instead of working directly with populations of potential solutions and handling the constraints using penalty functions or repairs, they proposed an indirect approach in which the task of balancing optimization and constraint satisfaction is shared between a greedy heuristic and the GA. Individuals are represented by permutations of the available nurses and the heuristic is used to build schedules by allocating the nurses to their shifts in the given order. Memetic algorithms [63,71], which are viewed as hybrid GA, are a population-based approach for heuristic search in optimization problems. Basically, they combine local search heuristics with crossover operators. Burke et al. [59] described a memetic algorithm that incorporates TS into a GA, using steepest descent for each individual. The results reported for the NRP are better than those obtained by a hybrid TS approach by Burke et al. [20]. This work has gone further in combining hybrid TS with evolutionary approaches.

There has been some use of simulated annealing techniques for the NRP. For example, Thompson [80] presented a SA heuristic for shift-scheduling using non-continuously available employees.

3.5. AI approaches

AI techniques have been used to solve NRPs modelled as a CSP. Chun et al. [28] modeled the NRP as a CSP which was solved by a combined approach of look-ahead and intelligent scoring which determines which nurse is to be scheduled next and which shift satisfies most of the soft constraints.

Abdennadher and Schlenker [1,2] adopt a PCSP model for the NRP. INTERDIP, which is their prototype system, supports semi-automatic creation of duty rosters and imitates certain aspects of manual planning to improve on the theoretical complexity of the problem, using a constraint package based on CHIP. The package includes linear equations, constraints over finite domains and boolean constraints.

Meyer auf'm Hofe [58] modeled the NRP as a HCSP, where legal regulations are hard constraints and nurses' preferences are usually lower-level soft constraints. Meyer auf'm Hofe [57] reported a commercial system ORBIS which models the NRP as a HCSP with fuzzy constraints and inferred control strategies. ORBIS uses a B&B algorithm with constraint propagation and variable/value-ordering techniques to solve problems involving 250–1200 variables withon few minutes.

Constraint logic programming languages have the advantage of describing constraint logic easily. Darmoni et al. [30] presented a non-cyclic scheduling system, namely Horoplan, whose algorithm is a constraint-based artificial intelligence approach implemented with Charme, which is a constraint-based programming language. Okada [67] discussed an approach, which takes advantage of the declarative ability of Prolog language for the description of constraints, for incorporating the constraints to generalize the NRP. Scott and Simpson [75] combined constraint logic programming with case-based reasoning to reduce the search spaces further.

As a commercial constraint-based package for the powerful C++ programming language, ILOG SOLVER has been widely used to solve the NRP, with the help of heuristic techniques (see [27,37,50,51,85]). It should be noted that Cheng et al. [27] used redundant modeling which increases constraint propagation through cooperation among different models for the same problem via channeling constraints.

Knowledge-based search approaches have also been used to solve the NRP by Lukman et al. [54] and Chen and Yeung [25].

4. On the evaluation of solution approaches

Although evaluation functions (penalty functions, target functions) can be used to evaluate particular algorithms, comparison, generally, between algorithms is very difficult. This is especially so when problem descriptions and models vary as widely as they do for the NRP and when the methods developed for their solution can be diverse. Comparison is further hindered by the lack of published experimental data and code. Some comparison among certain algorithms or hybrids algorithms can be made in certain specific contexts, usually by authors reporting specialized comparisons themselves, but there is still no complete and systematic way to evaluate all methods of solution to the range of NRPs described in the previous sections.

We give here an example to illustrate the difficulty of evaluation across problems. Because the trend has been to use meta-heuristics, we will first cite the experience of Burke et al. [20] and Burke and Cowling [21]. The typical problem size for the NRP here consists of 20 personnel per ward, six shift types and thirty types of active constraints per person over a planning period of 28 days. Burke et al. [20] solve the NRP with steepest descent method and variants of TS method, such as basic TS method, TS + diversification and TS + greedy shuffling. The result shows that TS algorithm is better than steepest descent and the hybrid TS is the best. Burke and Cowling [21] show that the TS heuristics can be made effective, especially for smaller rostering problems and show that new memetic approaches for the given problem are more robust than TS algorithms, at the expense of requiring longer solution times. An evaluation function and generation time are used to compare different algorithms. Also, a GA approach was attempted. For hybrid TS, they obtained better quality results than for the case of automating the rostering process directly where calculation times were acceptable. However, for GA, the calculation time was a hundred times that for hybrid TS while the results were of the same quality. In the case of memetic algorithms, the calculation time was ten times as long as GA, and the solutions about 10% better. However, they were less dependant on initialization and parameter changes. This is reported in [15].

Jackson et al. [41] experimented with three iterative improvement approaches along with four methods for generating an initial solution. The four methods used to construct initial solutions are a manual method, an iterative sampling method, a greedy constructive method, and a random-greedy constructive method. The three iterative improvement approaches are: random iterative improvement, HC iterative improvement, and TS. Two types of comparisons are performed: the comparison based on the cost obtained from iterative sampling and computation time comparison. A comparison of labour cost and fairness cost is also performed. Experiments showed that the simple randomized greedy algorithm is able to generate good schedules using very little computational effort.

Aickelin [6] and Dowsland [32] solve the same problem with genetic method, tabu method and

XPRESS MP respectively. The three methods are compared in terms of solution quality, robustness and ease of including possible future expansions of the problem. Aickelin also compares various types of GA. Six typical GAs running with various strategies are compared. Aickelin and White [7] describes, further, a comparison algorithm specifically for the type of GA models studied.

These examples highlight the specificity of the solution approaches. While different approaches performance vary from problem to problem, the quality or results can be measured by a number of different quality measures, depending again on the needs of the particular problem.

The evaluation of the various techniques could be facilitated by the availability of benchmark problems for the various basic models for the NRP. For example, the kind of benchmarks (Solomon's test cases) that can be found for the vehicle routing problem can be useful for basic comparisons of algorithms on certain agreed-upon sets of problems/data.

5. Conclusion and future work

In this work, we have provided the modeling and solution methodologies for the NRP in the form of a bibliographic survey of work done in the past decades. In the different application contexts, the NRP can be modeled as an optimization problem or as a decision problem.

Optimization approaches can lead to optimal solutions although computational time can restrict the size of any NRP. For many instances of the NRP, however, it is difficult to incorporate every hard/soft constraint, and this has led to modeling NRPs as decision problems. Further, purely optimization solutions can be costly in computer time, whereas hospitals are dynamic environments for which simpler and more adaptive solutions based on flexible algorithms or user intuition can be more valuable.

Earlier scheduling policies such as cyclic scheduling were aimed at reducing the work of nurse officers in producing duty rosters manually. Straightforward automation of cyclic scheduling or "heuristic" approaches based on scheduling policy unavoidably restricted the nurses' preferences and thus the quality of health care [69]. However, recently, heuristic approaches have had the advantage of exploiting faster computing speed and thus providing more meaningful solutions. More recently, meta-heuristics have led the way in producing near-optimal solutions for hard problems with relatively less computational effort. They could be used in solving the NRP as a CSOP, for example. Heuristics and meta-heuristics have been much researched and applied to the NRP recently and have met with varied success. This is not unlike the record of such applications in other areas of research, including general scheduling problems, where we find, for example, a particular meta-heuristic working well for a problem in terms of computational time but providing lower quality solutions than other meta-heuristic approaches. The strength of a particular technique varies, depending on the problem type. In the applications of such techniques, however, we find that meta-heuristics have been easily applied to NRPs for the problems studied. In the context of TS, for example, a move is naturally definable as an assigned shift type. Similarly in GA and memetic algorithms, the notions of crossover and mutation are easily implemented for those problems studied. TS and its hybridizations seem to be well suited for a number of applications while GA and memetic applications can be too time consuming in practice.

Due to the number of constraints involved in problems, NRPs are naturally modeled as CSPs or their variants. Combinations of systematical tree search approaches (for example, BT and B&B) with constraint solving techniques or other heuristic variable- and value-ordering strategies have been successfully applied. A variety of constraint logic programming languages and constraint-based packages for conventional programming languages are available to facilitate the description of problems and constraints. This has proven to be a popular approach since the effort at solving NRPs can be considerably reduced by using such tools. Certainly, the number of constraints that can be practicably handled by such tools surpasses many of the other techniques. These techniques have proven to be effective in a number of applications.

Hybrid algorithms, which are the particular combinations of AI techniques with the traditional optimization methods, such as IP, are promising. This is somewhat different from the situations where hybrid are combinations of purely AI techniques. We have seen that has been some success when such methods have been applied.

In view of the difficulty in evaluating solutions to NRPs, a complete benchmark database on certain standard NRPs will help in comparisons of the various algorithms applied to NRPs and provide for greater efficiency in developing better solutions for this class of problems.

Acknowledgements

The authors wish to thank the anonymous referees whose comments have contributed to improvements in this work. They wish also to express their indebtedness to authors who's works this study has drawn upon.

References

[1] S. Abdennadher, H. Schlenker, INTERDIP—an interactive constraint based nurse scheduler, in: PACLP-99. Available from <http://www.pms.informatik.unimuenchen.de/~interdip>, 1999.
[2] S. Abdennadher, H. Schlenker, Nurse scheduling using constraint logic programming, AAAI/IAAI, 1999, pp. 838–843.
[3] H. Ahuja, R. Sheppard, Computerized nurse scheduling, Industrial Engineering 7 (10) (1975) 24–29.
[4] U. Aickelin, K. Dowsland, An indirect genetic algorithm for a nurse scheduling problem, Computers and Operations Research, submitted for publication. Available from <www.inf.brad.ac.uk/~uaickeli/papers/02COR_indirect.pdf>, 2000, p. 26.
[5] U. Aickelin, K. Dowsland, Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem, Journal of Scheduling 3 (3) (2001) 139–153.
[6] U. Aickelin, Genetic algorithms for multiple-choice problems, Ph.D. thesis, University of Wales, Swansea, 1999.
[7] U. Aickelin, P. White, Building better nurse scheduling algorithms, Annals of OR, submitted for publication.
[8] M. Anzai, Y. Miura, Computer program for quick work scheduling of nursing staff, Medical Information 12 (1) (1987) 43–52.
[9] J.L. Arthur, A. Ravidran, A multiple objective nurse scheduling model, AIIE Transactions 13 (1981) 55–60.
[10] N. Balakrishnan, R.T. Wong, A network model for the rotating workforce scheduling problem, Networks 20 (1990) 25–41.
[11] K.R. Baker, Scheduling a full-time workforce to meet cyclic staffing requirements, Management Science 20 (1974) 1561–1568.
[12] R. Bartak, Constraint programming: In pursuit of the holy grail, in: Proceedings of WDS99, Prague, 1999, p. 10.
[13] J.J. Bartholdi, J.B. Orlin, H.D. Ratfill, Cyclic scheduling via integer programs with circular ones, Operations Research 28 (1980) 1074–1085.
[14] P.C. Bell, G. Hay, Y. Liang, A visual interactive decision support system for workforce (nurse) scheduling, INFOR 24 (1986) 134–145.
[15] G.V. Berghe, Nurse rostering—a practical case study, undated, School of Computer Science and IT, University of Nottingham. Available from <http://www.cs.nott.ac.uk>.
[16] I. Berrada, J.A. Ferland, P. Michelon, A multi-objective approach to nurse scheduling with both hard and soft constraints, Socio-Economic Planning Science 30 (20) (1996) 183–193.
[17] J.E. Beasley, Advances in Linear and Integer Programming, Oxford University Press, 1996.
[18] A. Borning, R. Duisberg, B. Freeman-Benson, K. Kramer, M. Woolf, Constraint hierarchies, in: Proceedings of ACM Conference on Object Oriented Programming Systems, Languages and Applications, ACM, 1987, pp. 48–60.
[19] A. Borning, B. Freeman-Benson, M. Wilson, Constraint hierarchies, LISP and Symbolic Computation 5 (3) (1992) 223–270.
[20] E. Burke, P.D. Causmaecker, G.V. Berghe, A hybrid tabu search algorithm for the nurse rostering problem, in: X. Yao et al. (Eds.), SEAL'98, LNCS, vol. 1585, Springer, Berlin, Heidelberg, 1999, pp. 187–194.
[21] E. Burke, P. Cowling, P.D. Causmaecker, G.V. Berghe, A memetic approach to the nurse rostering problem, Applied Intelligence 15 (2001) 199–214.
[22] E.K. Burke, P.D. Causrnaecker, S. Petrovic, G.V. Berghe, Fitness evaluation for nurse scheduling problems, in: Proceedings of the Congress on Evolutionary Computation—CEC, Seoul, Korea, 2001, pp. 1139–1146.
[23] R.N. Burns, Manpower scheduling with variable demands and alternate weekends off, INFOR 16 (1978) 101–111.
[24] R.N. Burns, G.J. Koop, A modular approach to optimal multiple-shift manpower scheduling, Operations Research 35 (1987) 100–110.
[25] J.G. Chen, T.W. Yeung, Hybrid expert-system approach to nurse scheduling, Computers in Nursing 11 (1993) 183–190.
[26] B.M.W. Cheng, J.H.M. Lee, J.C.K. Wu, A constraint-based nurse rostering system using a redundant modeling approach. 8th International Conference on Tools with Artificial Intelligence (ICTAI '96), November 16–19, 1996, pp. 140–148.
[27] B.M.W. Cheng, J.H.M. Lee, J.C.K. Wu, A nurse rostering system using constraint programming and redundant

modeling, IEEE Transactions in Information Technology in Biomedicine 1 (1) (1997) 44–54.
[28] A.H.W. Chun, S.H.C. Chan, G.P.S. Lam, F.M.F. Tsang, J. Wong, D.W.M.Yeung, Nurse rostering at the hospital authority of Hong Kong, AAAI/IAAI 2000, pp. 951–956.
[29] A. Colmerauer, An introduction to Prolog III. Communications of the ACM, 1990, pp. 69–90.
[30] S.J. Darmoni, A. Fajner, N. Mahe, A. Leforetier, M. Vondracek, O. Stelian, M. Baldenweck, Horoplan: Computer-assisted nurse scheduling using constraint-based programming. Available from <www.chu-rouen.fr/dsii/publi/plao.html>, 2000, p. 16.
[31] M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun, T. Graf, F. Berthier, The constraint logic programming language CHIP, in: Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88), Tokyo, Japan, 1988, pp. 693–702.
[32] K.A. Dowsland, Nurse scheduling with tabu search and strategic oscillation, European Journal of Operational Research 106 (1998) 393–407.
[33] S.M.A. Frances, Implementing a program of cyclical scheduling of nursing personnel, Hospitals 40 (1966) 108–125.
[34] E.C. Freuder, R.J. Wallace, Partial constraint satisfaction, Artificial Intelligence 58 (1–3) (1992) 21–70.
[35] F. Glover, C. Mcmillan, The general employee scheduling problem: An integration of management science and artificial intelligence, Computers and Operations Research 13 (1986) 563–573.
[36] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, 1997.
[37] K. Heus, G. Weil, Constraint programming a nurse scheduling application, in: Proceedings of the Second International Conference on the Practical Application of Constraint Technology, 1996, pp. 115–127.
[38] J.P. Howell, Cyclical scheduling of nursing personnel, Hospitals 40 (1966) 77–85.
[39] R. Hung, Hospital nurse scheduling, Journal of Nursing Administration 25 (7/8) (1995) 21–23.
[40] ILOG, ILOG SOLVER 4.4 Reference Manual, 1999.
[41] W.K. Jackson, W.S. Havens, H. Dollard, Staff scheduling: A simple approach that worked. Available from <http://www.cs.sfu.ca/research/groups/ISL/papers/jackson-staff/node1.html>, 1997.
[42] J. Jaffar, M.J. Maher, Constraint logic programming: A survey, Journal of Logic Programming 19/20 (1994) 503–581.
[43] J. Jaffar, S. Michaylov, P.J. Stuckey, R.H.C. Yap, The CLP (R) language and system, ACM Transactions on Programming Languages and Systems 14 (3) (1992) 339–395.
[44] B. Jaumard, F. Semet, T. Vovor, A generalized linear programming model for nurse scheduling, European Journal of Operational Research 107 (1998) 1–18.
[45] R.C. Jelinek, J.A. Kavois, Nurse staffing and scheduling: Past solutions and future directions, Journal of the Society for Health Systems 3 (1992) 75–82.
[46] A. Jan, M. Yamamoto, A. Ohuchi, Evolutionary algorithms for nurse scheduling problem. in: Proceedings of the 2000 Congress on Evolutionary Computation CEC00, IEEE Press, 2000, pp. 196–203. Available from <http://www.lania.mx/~ccoello/EMOO/jan00.ps.gz> 8 pp.
[47] M.M. Kostreva, K.S.B. Jennings, Nurse scheduling on a microcomputer, Computers and Operational Research 18 (8) (1991) 731–739.
[48] L. Kragelund, T. Kabel, Employee timetabling, Master's thesis in computer science, University of Aarhus, 1998.
[49] V. Kumar, Algorithms for constraint satisfaction problems: A survey, AI Magazine 13 (1) (1992) 32–44.
[50] S. Kusumoto, Nurse scheduling system using ILOG Solver, in: Proceedings of the Second ILOG Solver and Scheduler Users Conference, Paris, 1996, p. 6. Available from <www.ilog.com/products/optimization/tech/custpapers/nec.pdf>.
[51] J.M. Lazaro, P. Aristondo, Using Solver for nurse scheduling, ILOG Solver and ILOG Schedule, First International Users' Conference Proceedings, 1995, p. 10.
[52] Linear programming frequently asked questions, Optimization Technology Center of Northwestern University and Argonne National Laboratory. Available from <http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html>.
[53] E.W. Lawler, D.E. Wood, Branch-and-bound methods: A survey, Operations Research 14 (1966) 699–719.
[54] D. Lukman, J.H. May, L.J. Shuman, H.B. Wolfe, Knowledge-based schedule formulation and maintenance under uncertainty, Journal of Society of Health Systems 2 (2) (1991) 42–64.
[55] C. Mailer-Rothe, H.B. Wolfe, Cyclical scheduling and allocation of nursing staff, Socio-Economic Planning Science 7 (1973) 471–487.
[56] P.M. Marchionno, Modified cyclical scheduling: A practical approach, Nursing Management 18 (1987) 60–66.
[57] H. Meyer auf'm Hofe, Nurse rostering as constraint satisfaction with fuzzy constraints and inferred control strategies, in: E.C. Freuder, R.J. Wallace (Eds.), Constraint Programming and Large Scale Discrete Optimization, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 57, DIMACS, 2001, pp. 67–99.
[58] H. Meyer auf'm Hofe, ConPlan/SIEDAplan: Personal assignment as a problem of hierarchical constraint satisfaction, in: Proceedings of the Third International Conference on the Practical Application of Constraint Technology, 1997, pp. 257–272.
[59] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolutionary Programs, third ed., Springer-Verlag, 1997.
[60] H. Millar, M. Kiragu, Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming, European Journal of Operational Research 104 (3) (1992) 582–592.
[61] H.E. Miller, P. William, J.R. Gustave, Nurse scheduling using mathematical programming, Operations Research 24 (5) (1976) 857–870.

[62] G. Monroe, Scheduling manpower for service operations, Industrial Engineering 2 (8) (1970) 10–17.
[63] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Technical Report C3P 826, Pasadena, CA, 1989.
[64] A.A. Musa, U. Saxena, Scheduling nurses using goal-programming techniques, IIE Transactions 16 (1984) 216–221.
[65] Nonlinear programming frequently asked questions, Optimization Technology Center of Northwestern University and Argonne National Laboratory. Available from <http://www-unix.mcs.anl.gov/otc/Guide/faq/nonlinearprogramming-faq.html>.
[66] K. Nonobe, T. Ibaraki, A tabu search approach to the constraint satisfaction problem as a general problem solver, European Journal of Operational Research 106 (1998) 599–623.
[67] M. Okada, An approach to the generalized nurse scheduling problem—generation of a declarative program to represent institution-specific knowledge, Computer and Biomedical Research 25 (5) (1992) 417–434.
[68] Mihoko Okada, Masahiko Okada, Prolog-based system for nursing staff scheduling implemented on a personal computer, Computer and Biomedical Research 21 (1988) 53–63.
[69] J.H. Oldenkamp, Quality in Fives: On the Analysis, Operationalization and Application of Nursing Schedule Quality, University of Groningen, 1996.
[70] I. Ozkarahan, J.E. Bailey, A multi-objective approach to nurse scheduling with both hard and soft constraints, IIE Transactions 20 (3) (1988) 306–316.
[71] N.J. Radcliffe, P.D. Surry, Formal memetic algorithms, in: T.C. Fogarty (Ed.), Evolutionary Computing: AISB Workshop, Berlin, Springer-Verlag, 1994, pp. 1–16.
[72] S.U. Randhawa, D. Sitompul, A heuristic based computerized nurse scheduling system, Computers and Operations Research 20 (8) (1983) 837–844.
[73] G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufman, San Mateo, CA, 1991.
[74] E.S. Rosenbloom, N.F. Goertzen, Cyclic nurse scheduling, European Journal of Operational Research 31 (1987) 19–23.
[75] S. Scott, R. Simpson, Case-bases Incorporating Scheduling Constraint Dimensions—Experiences in Nurse Rostering, EWCBR, 1998, pp. 392–401.
[76] D. Sitompul, S. Randhawa, Nurse scheduling models: A state-of-the-art review, Journal of the Society of Health Systems 2 (1990) 62–72.
[77] L.D. Smith, A. Wiggins, A computer-based nurse scheduling system, Computers and Operations Research 4 (1977) 195–212.
[78] Soft constraints in the nurse Rostering problem, Full description of all the constraint types in use in Plane System, February 6, 2001. Available from <http://www.cs.nott.ac.uk/~gvb/constraints.ps>.
[79] F.E. Tackling, Scheduling problems using integer programming, Master Thesis, University of Wales, Swansea, 1998.
[80] G.M. Thompson, A simulated annealing heuristic for shift-scheduling using non-continuously available employees, Computers and Operations Research 23 (1996) 275–288.
[81] C. Valouxis, E. Housos, Hybrid optimization techniques for the workshift and rest assignment of nursing personnel, Artificial Intelligence in Medicine 20 (2000) 155–175.
[82] P. Van Hentenryck, Constraint Satisfaction in Logic Programming, MIT Press, Cambridge, MA, 1989.
[83] D.M. Warner, J. Prawda, A mathematical programming model for scheduling nursing personnel in a hospital, Management Science 19 (4) (1972) 411–422.
[84] D.M. Warner, Scheduling nursing personnel according to nursing preference: A mathematical programming approach, Operations Research 24 (5) (1976) 842–856.
[85] G. Weil, K. Heus, P. Francois, M. Poujade, Constraint programming for nurse scheduling, IEEE Transactions in Engineering in Medicine and Biology (1995) 417–422.